Louis Williams

Due 2nd June 2017 – 11:59pm

IFB299 – Application Design and Development

personal portfolio

Table of Contents

[Release 1 3](#_Toc484806062)

[Login and Registration 3](#_Toc484806063)

[Organising stand ups and meetings 4](#_Toc484806064)

[Password Hashing 5](#_Toc484806065)

[User Stories 6](#_Toc484806066)

[Tests Cases 7](#_Toc484806067)

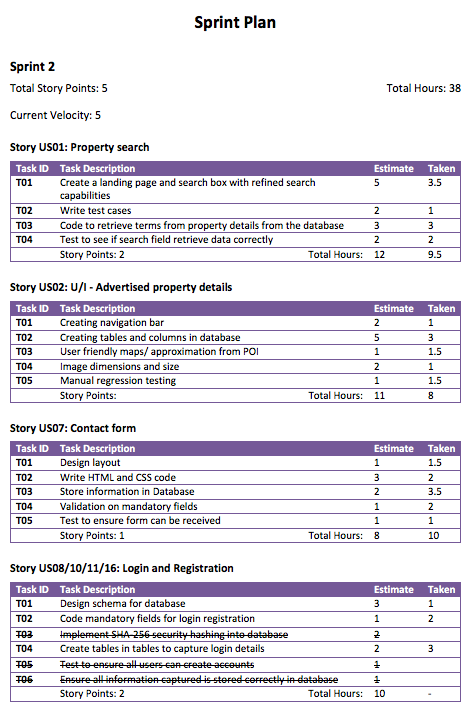
[Research on PHP and MySQL 8](#_Toc484806068)

[Database Requirements Specifications 9](#_Toc484806069)

[Re-formatted code of webpages to keep website constant 10](#_Toc484806070)

# Release 1

For sprint 1, our development team focused on creating the property data details, property search and login/registration page. I was tasked with ensuring the database and PHP scripts worked in order for our users to login and create accounts successfully. As seen in the table below, I wasn’t able to successfully create the correct PHP scripts and integrate them with the MySQL database. Resulting in tasks T03, T05 and T06 were all moved to Release 2.



## Login and Registration

For Sprint 1 I was tasked with ensuring users (i.e. home owners and customers) were able to sign up and create accounts, then login. This involved a lot of research into how to integrate PHP with HTML and structure tables in a MySQL database to store user’s details. I used bootstrap template to create a user friendly Graphical User Interface which complimented the company’s colours. For the most part I created a login and registration interface on the same page, which resulted in less development hours.

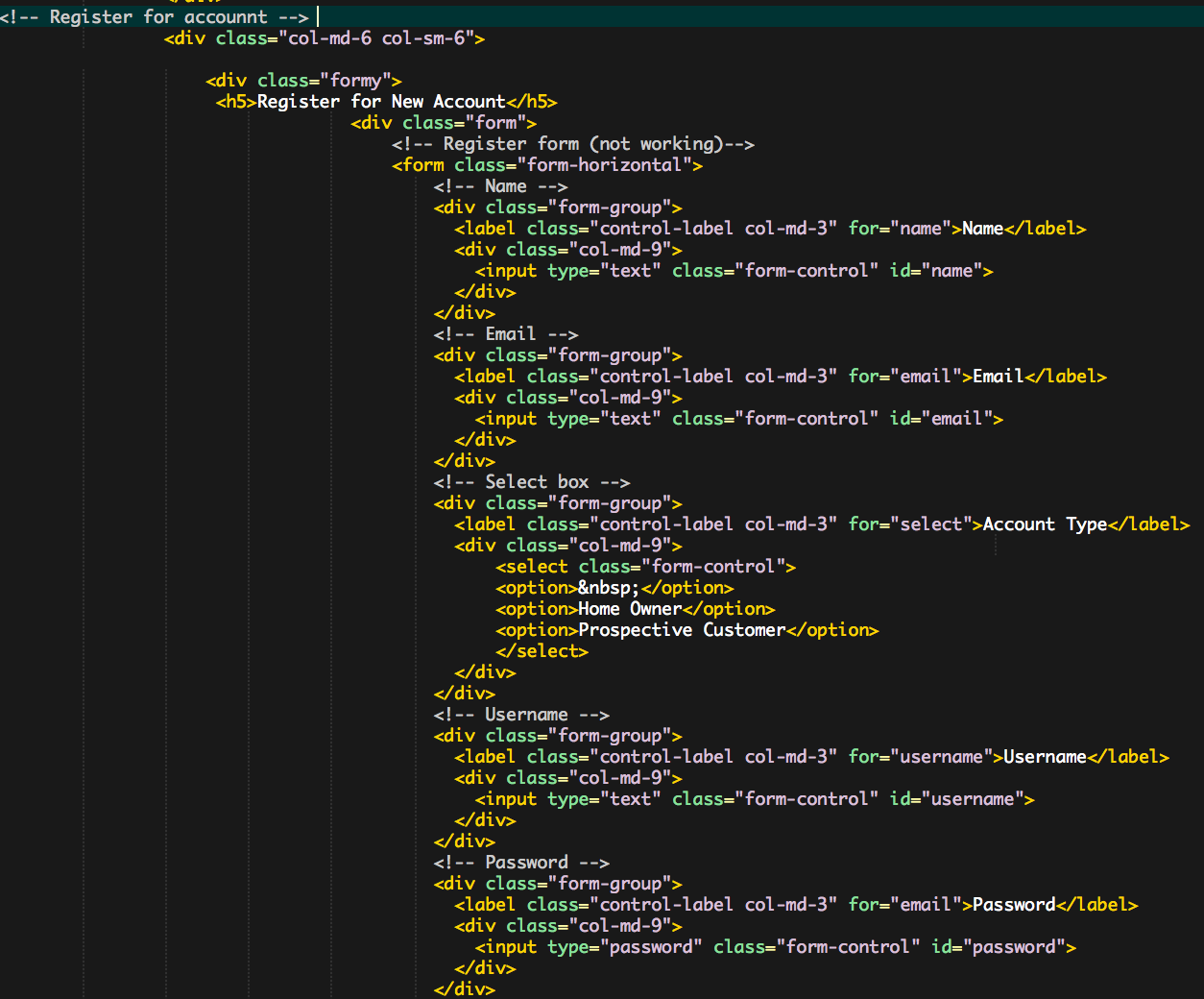
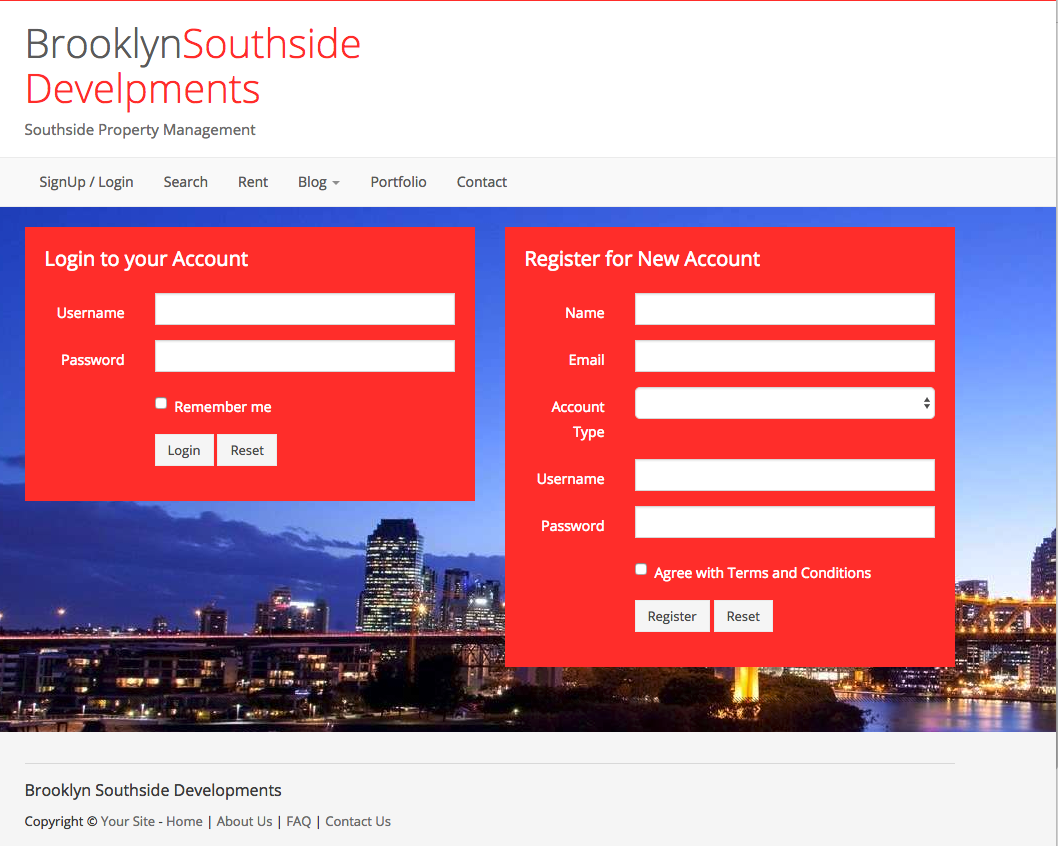


Figure 1 - Login and Registration output Figure 2 - Registration html code

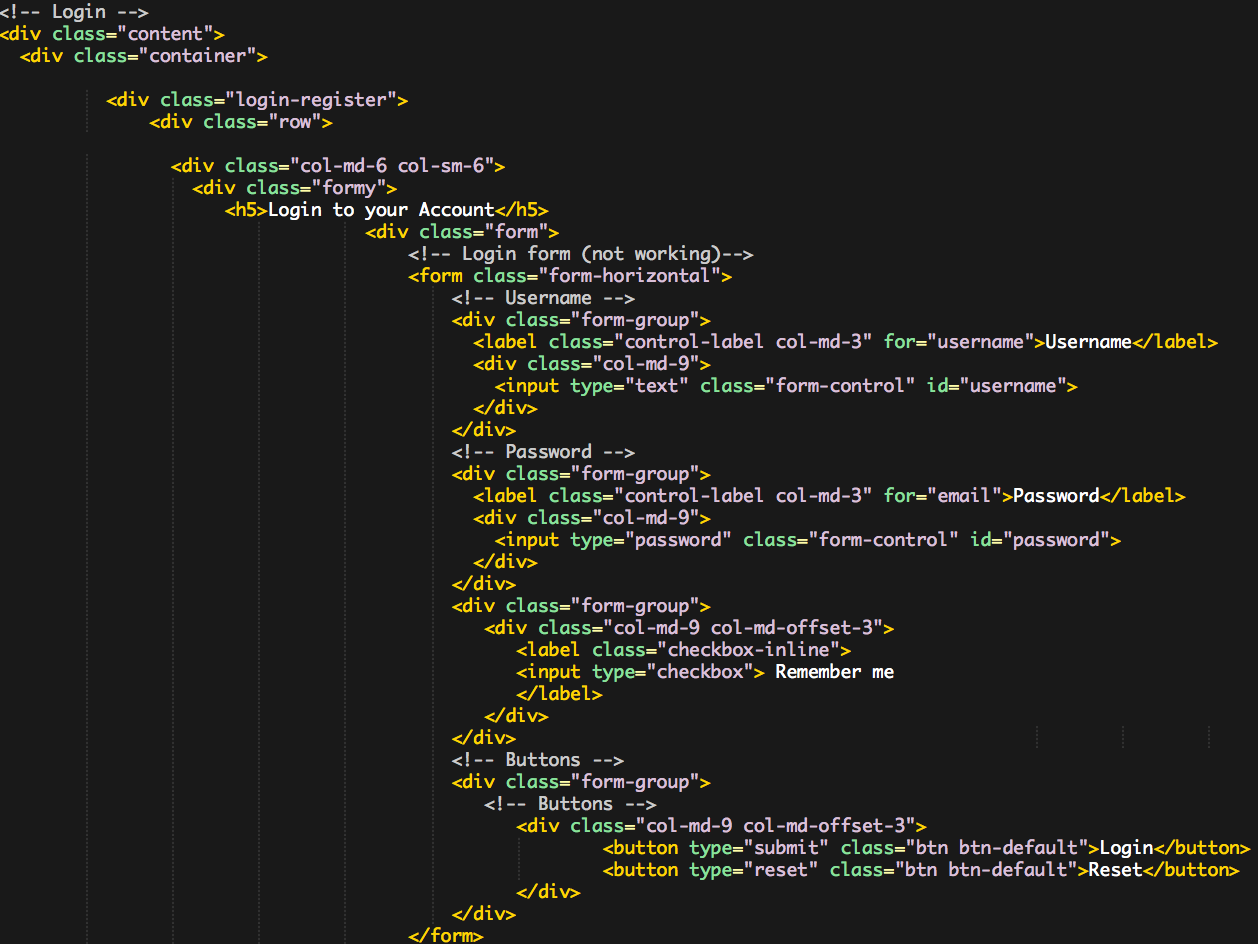
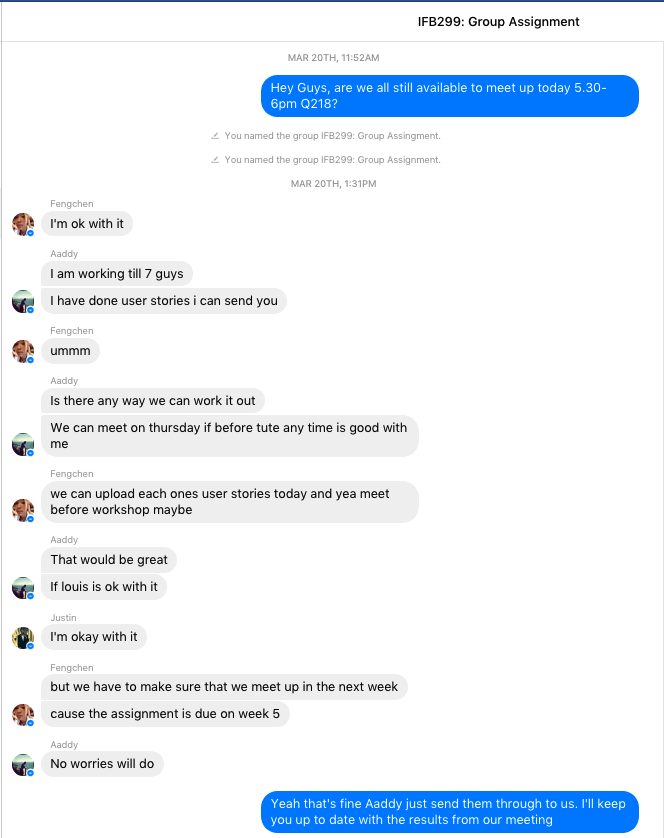
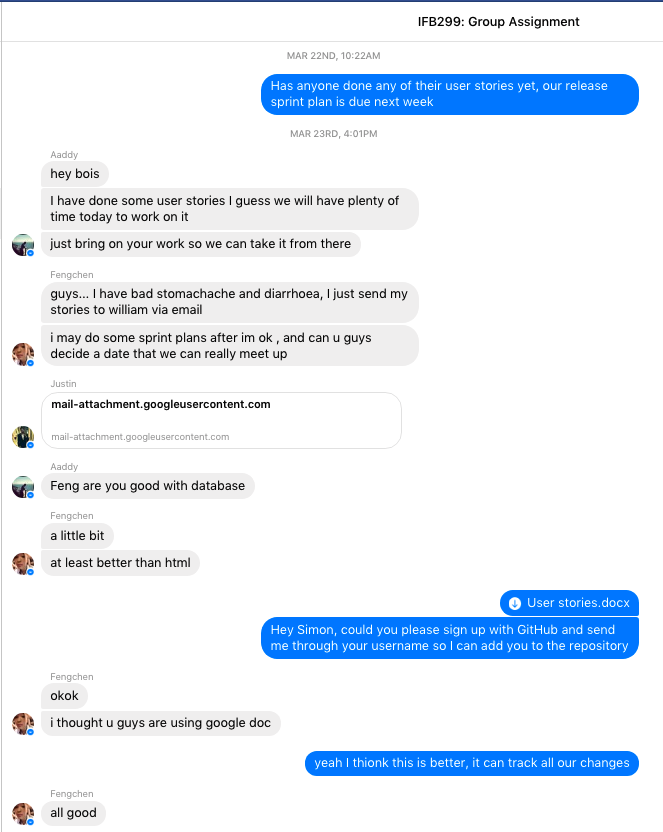


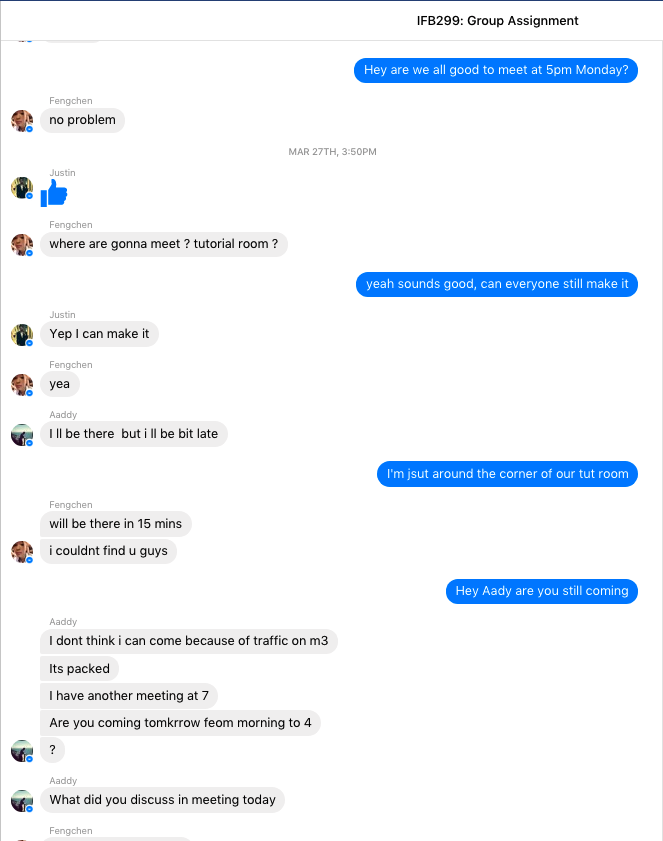
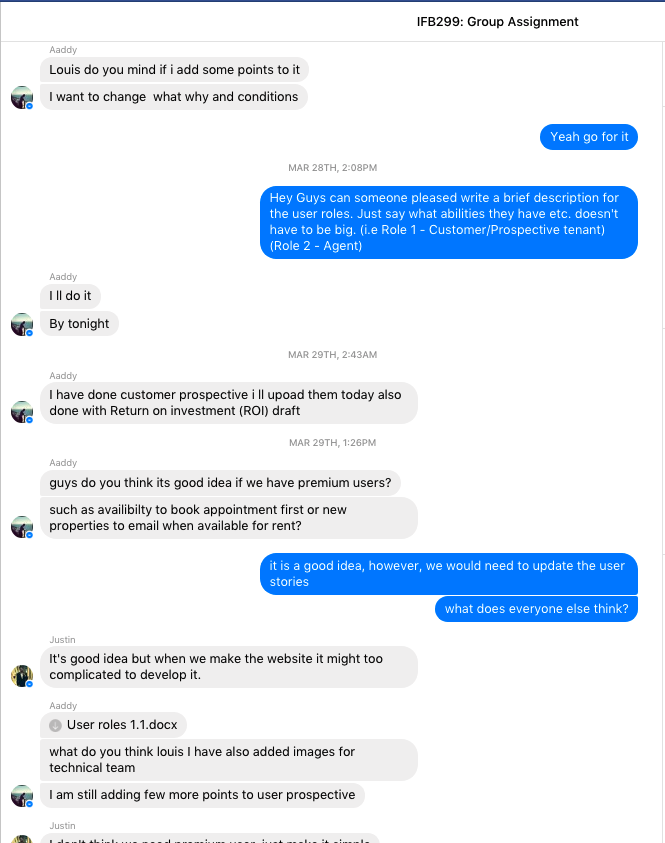
Figure 3 - Login html code

## Organising stand ups and meetings

Throughout the semester, I arranged meetings through Facebook so the team could communicate and decide what task were left. I tried getting the team in the habit of communicating through our Facebook group so we could find a time that suited everyone. It was also a way for each of us to give each other feedback on any documentation that had been completed.

We tried to have one meeting each week other than our tutorial to ensure the project was on track. Each time we would have a meeting we would discuss our feedback and upload the document to Facebook and on GitHub. I also used our group chat of way to delegate tasks that had not been completed yet to other members of the team.

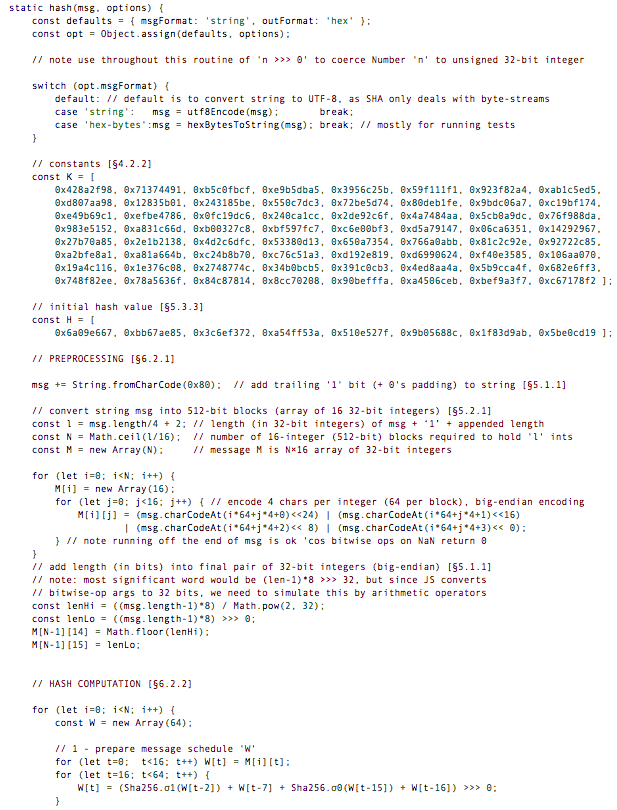
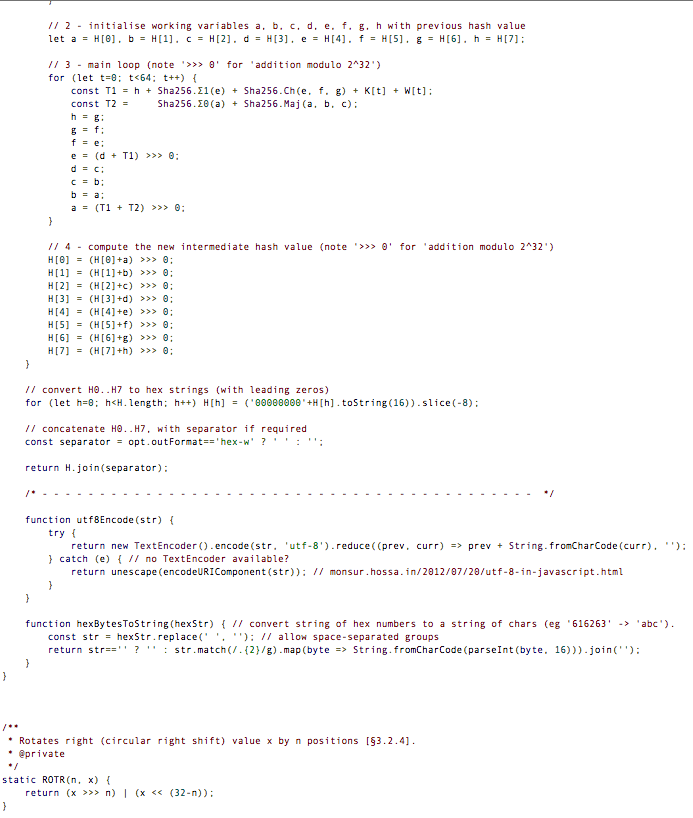
 

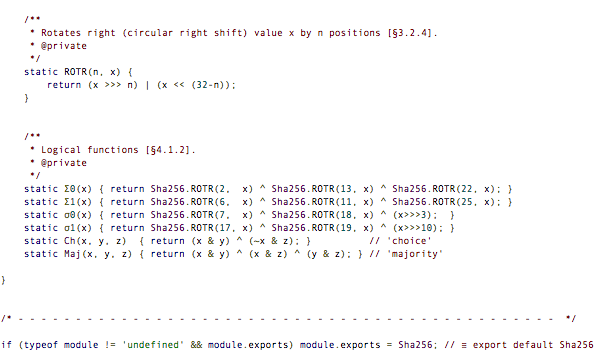
## Password Hashing

As task T03 was to use a Secure Hashing Algorithm (i.e. SHA-256) and implement that into the website for added security to store passwords. From the research I conducted, I found that SHA-256 is a cryptographic one way function that cannot be decrypted back. This involved hashing which is a type of algorithm that takes any size of data and turns it into a fixed length of data. This makes it ideal for a password validation, challenge hash authentication and other methods for hackers to penetrate the system. Once the hash is generated it creates and almost unique 256-bit (32-byte) signature for text. As an example, if were to take the string “Hello World”, it would generate the following hash:

**A591A6D40BF420404A011733CFB7B190D62C65BF0BCDA32B57B277D9AD9F146E**

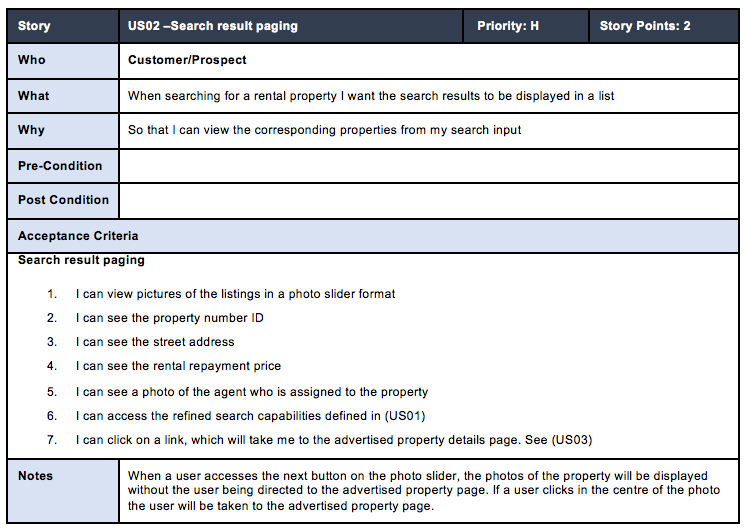
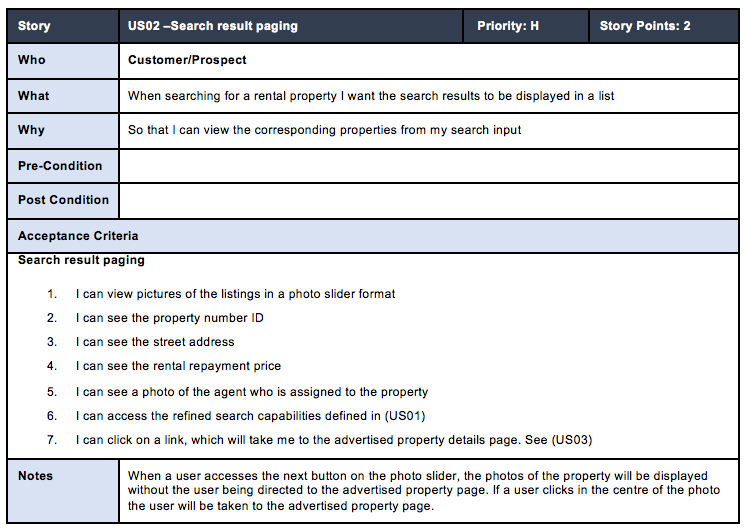
The part where I struggled was how to implement this algorithm into the database to safely secure all users passwords. I did find a piece of JavaScript that someone else had developed, however, I was not completely satisfied or sure if this was secure. We have moved this task into Release 2 as I needed more time to research into how to implement this algorithm correctly.



## User Stories

Before Sprint 1 had started I elected to create most of the user stories with inputs and feedback from my team members. In total, we had 17 user stories which followed the structured format I specified in the early stages of the project. This way we could better outline the acceptance criteria and help keep the user stories consistent. To get an idea of what the business requirements were, we first brainstormed some idea’s. This can be found in our git repository IFB299-Brooklyn-Southside-Developments/Documents/Brainstormngideas.gdoc. The process we followed to construct our user stories was, I would incorporate our ideas from our brainstorming session, then I would define the acceptance criteria. As a team, we would go through each story together to integrate everyone’s feedback. Below are some of the user stories I constructed:



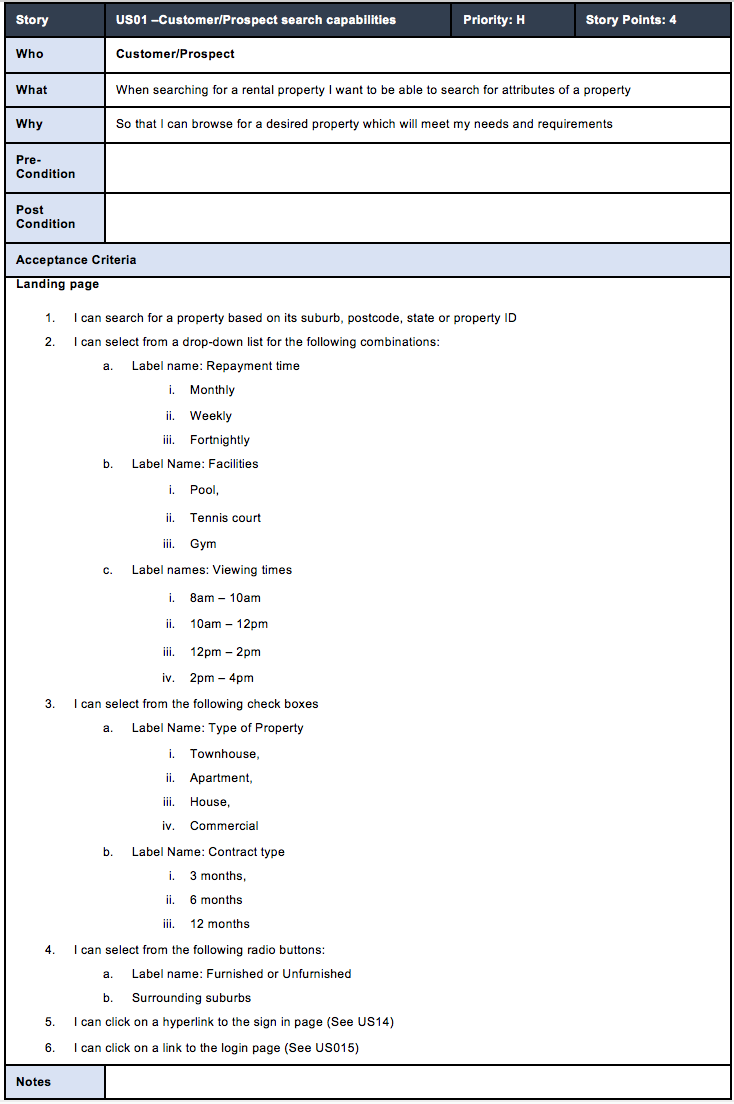
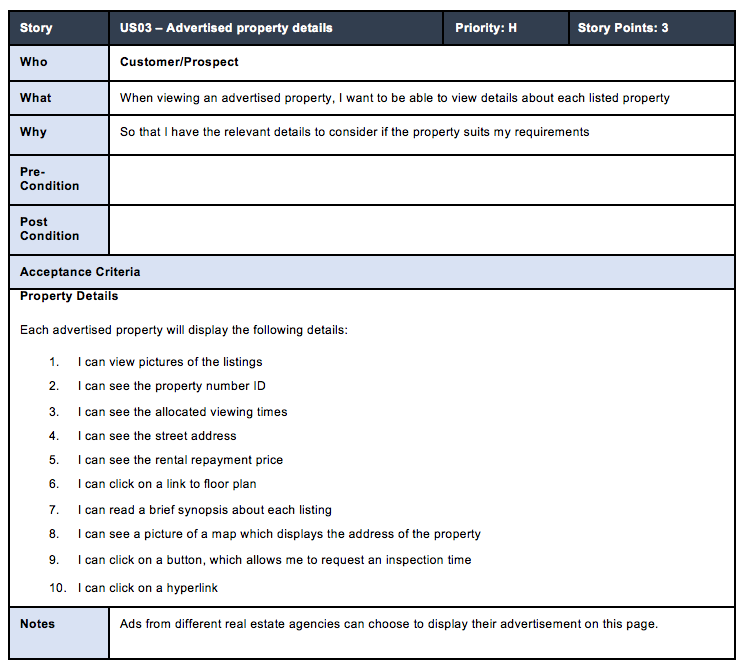
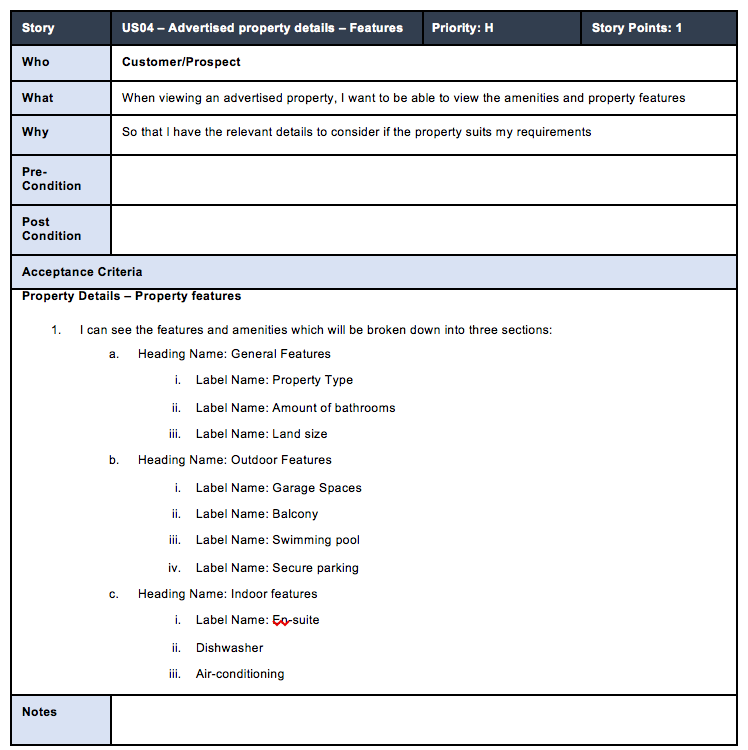


Figure 5 -Search result paging

Figure 4 – Customer Search capabilities

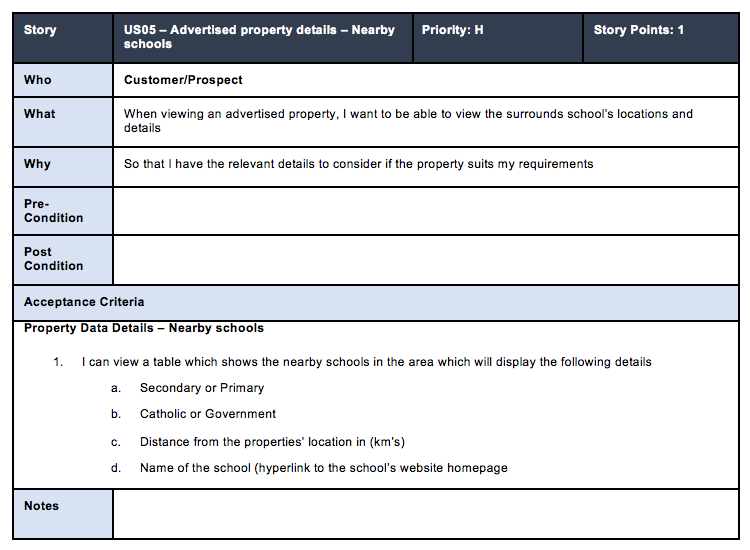


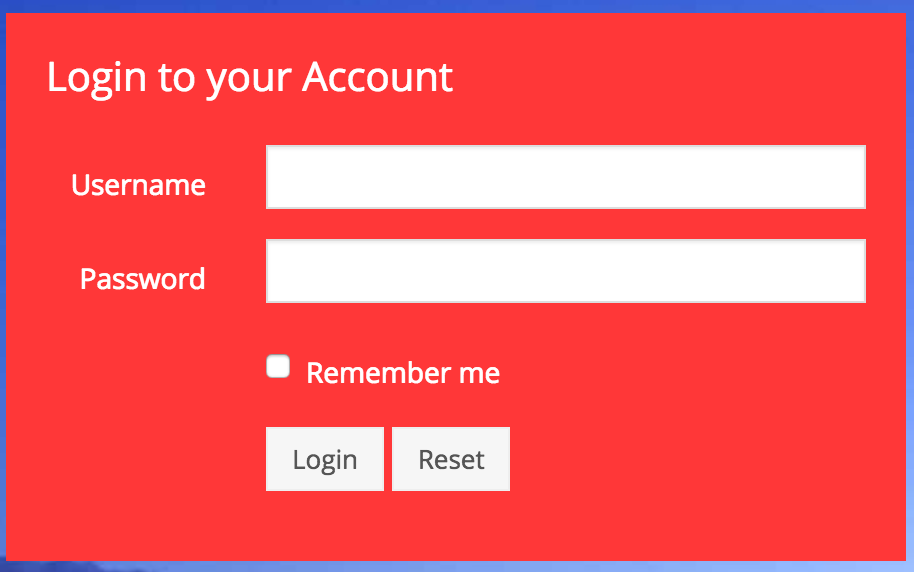
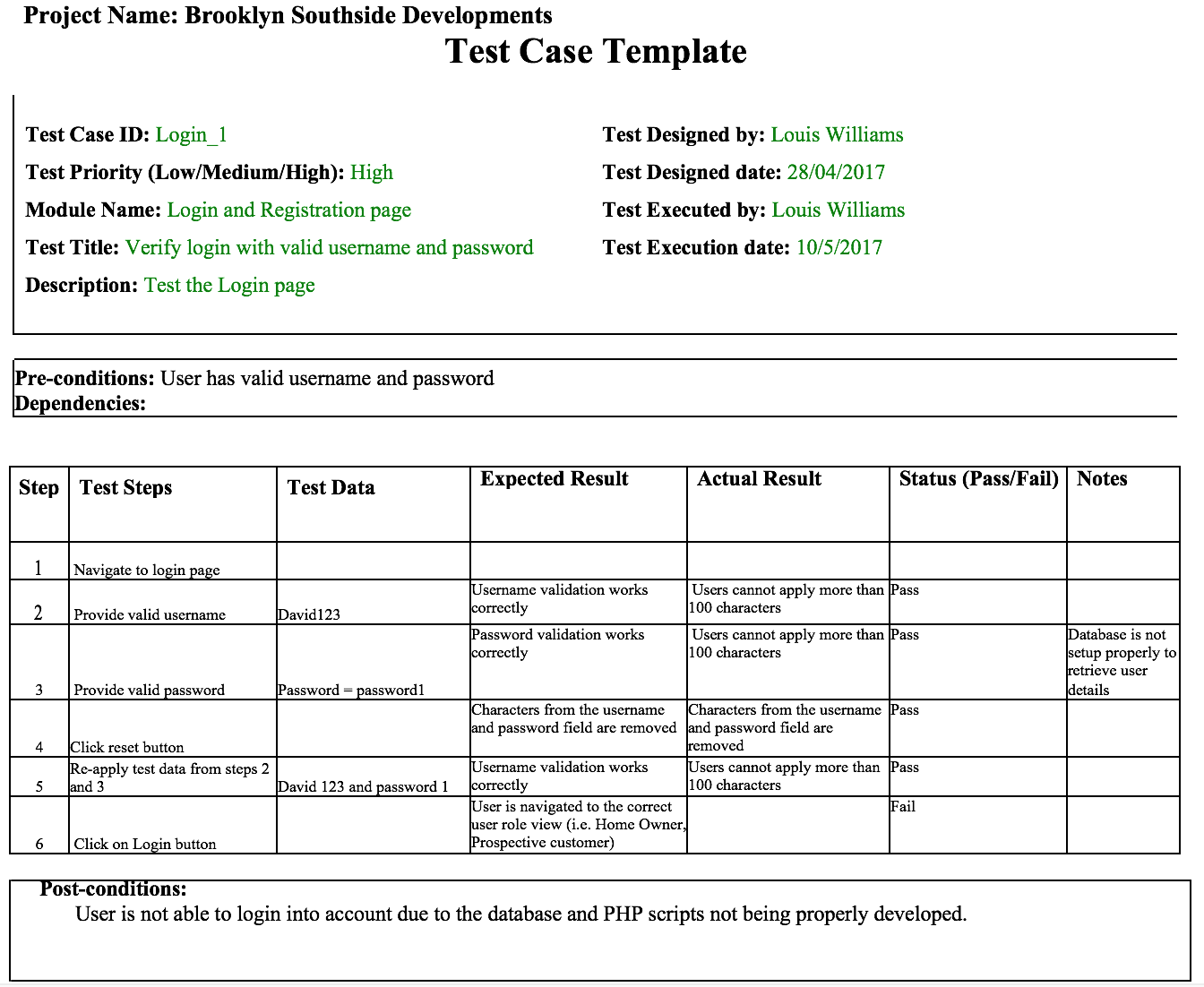
Figure 7 - Advertised property details - Features

Figure 6 - Advertised property details

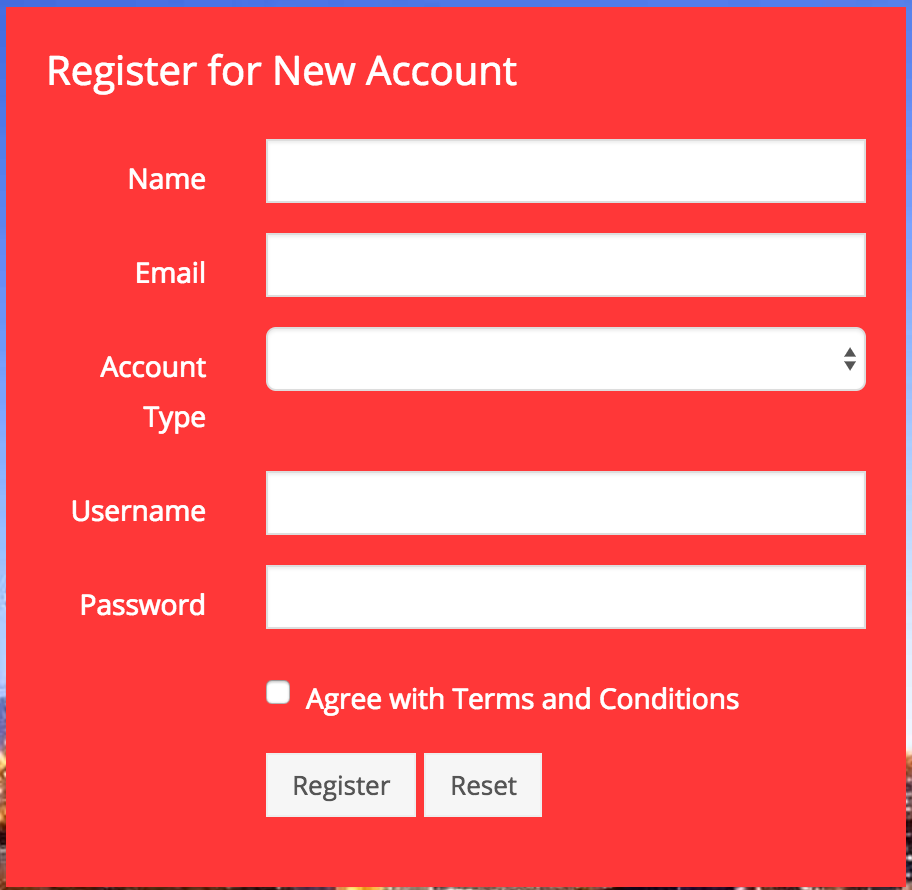
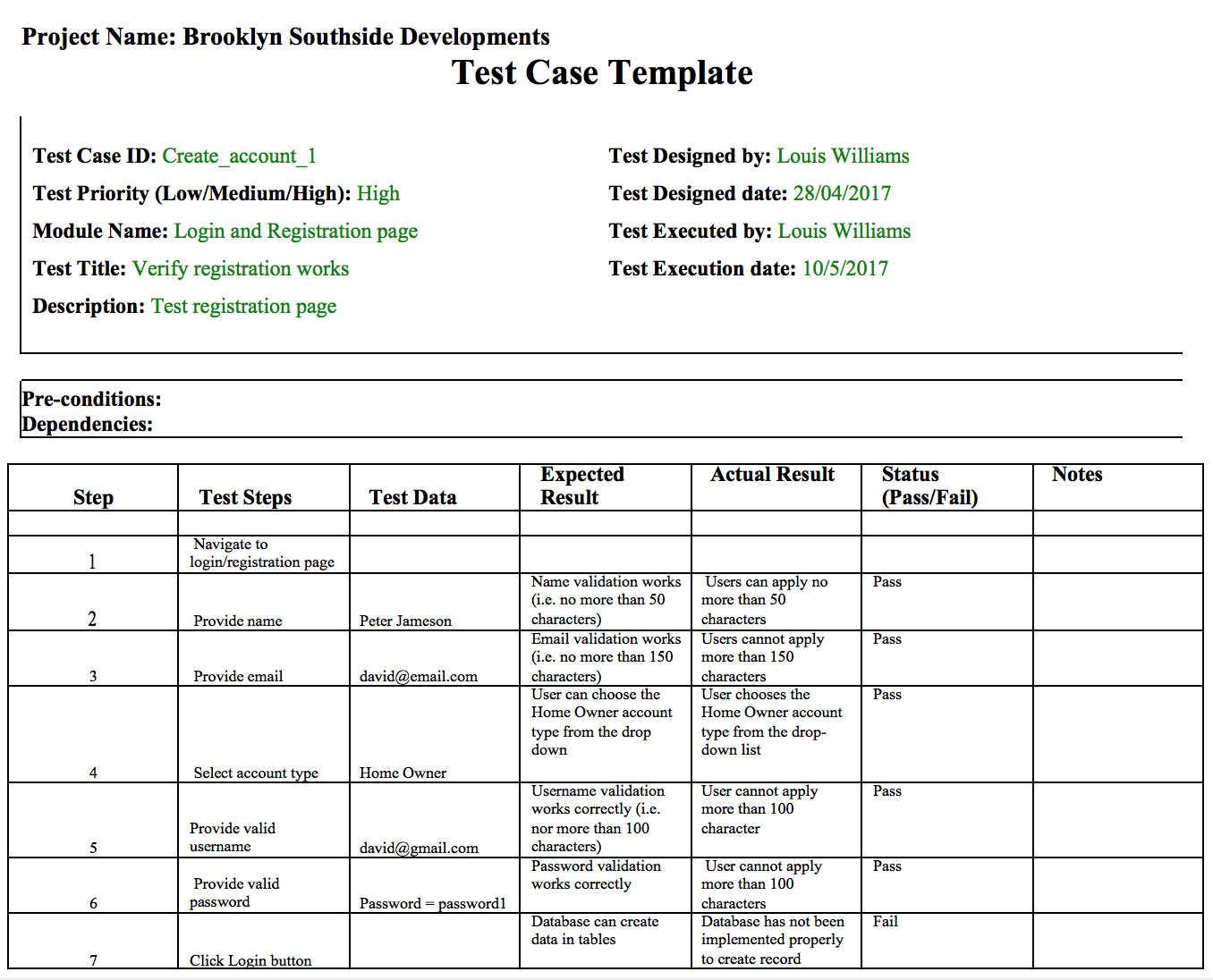
Figure 8 - Advertised Property details - Nearby schools

## Tests Cases

Below are two test cases I wrote to ensure the Login and Registration forms worked as defined in the acceptance criteria. The first one was constructed to ensure the max character lengths for the username and password fields work as intended. This test case also ensures that the reset button (which deletes characters in the username and password fields) works as indented. All these steps passed successfully, however, step 6 failed. Step 6 failed due to the database not working correctly. I created the tables and some of the PHP scripts but could never get to the stage to test it on my local machine. This was due to my lack of experience with databases and coding in general. This task was then moved to release 2 so the developers could spend more time on researching this functionality.

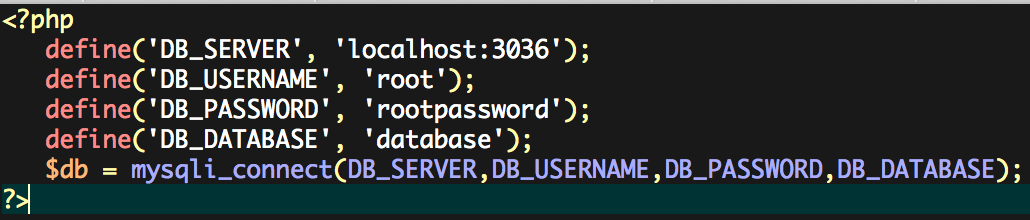
 

The test case seen below was developed to ensure the max character length, drop-down and buttons worked as intended. Most the testing steps passed, however, the step 7 failed. As listed above, I could never get the PHP scripts to work correctly and was having databases issues. This functionality was moved to Release 2 to allow the developers more time to research on how to implement this effectively.

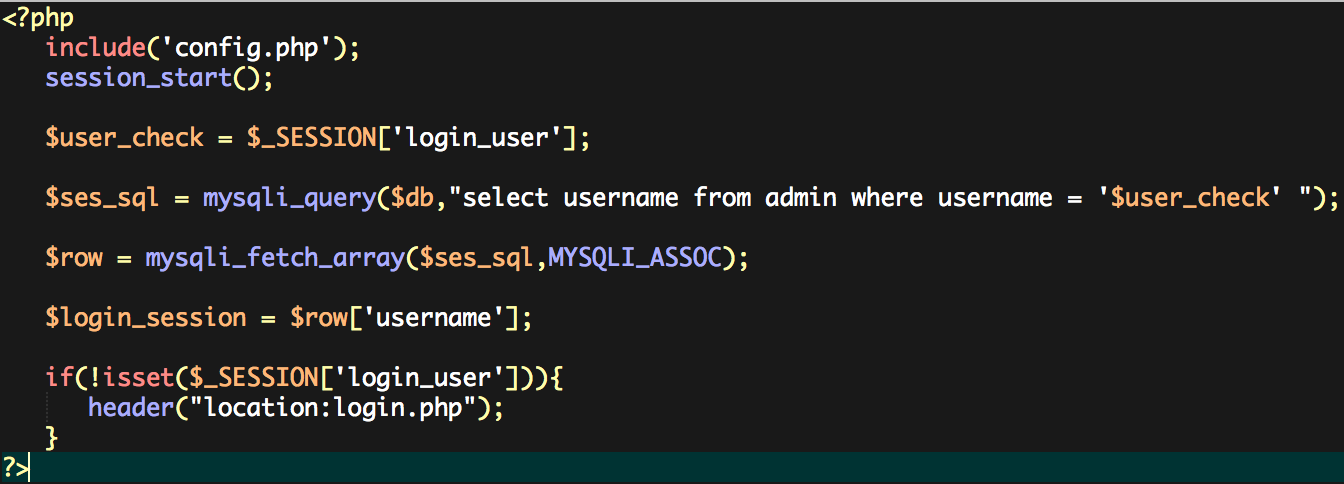
## Research on PHP and MySQL

I researched and watched many tutorials on PHP and MySQL to get and understanding of how implement these code bases into the website. From prior knowledge, I knew that PHP would be required to retrieve information from a database. For this I wrote a config.php file which connected to the MySQL database.

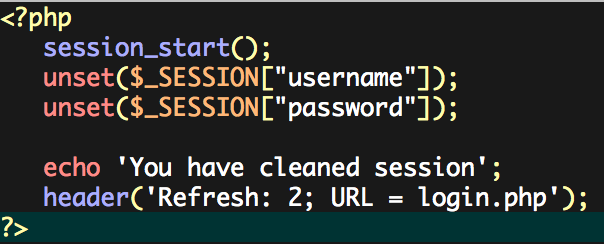


This tells MySQL that the web sever (the entity running the script) is on the local machine has a username of root, and has a password of rootpassword and database name of database.

A session is used to maintain a state between pages. Once the session it registers a unique session ID which is sent to the user via a cookie. PHP then creates a corresponding file on the server that can then keep track of any number of variables. Once the session is created, you can register any number of variables. The values of these variables are kept in the file on the server. If the session cookie lives, these variables will be available to any page within the same domain that wishes to access them.

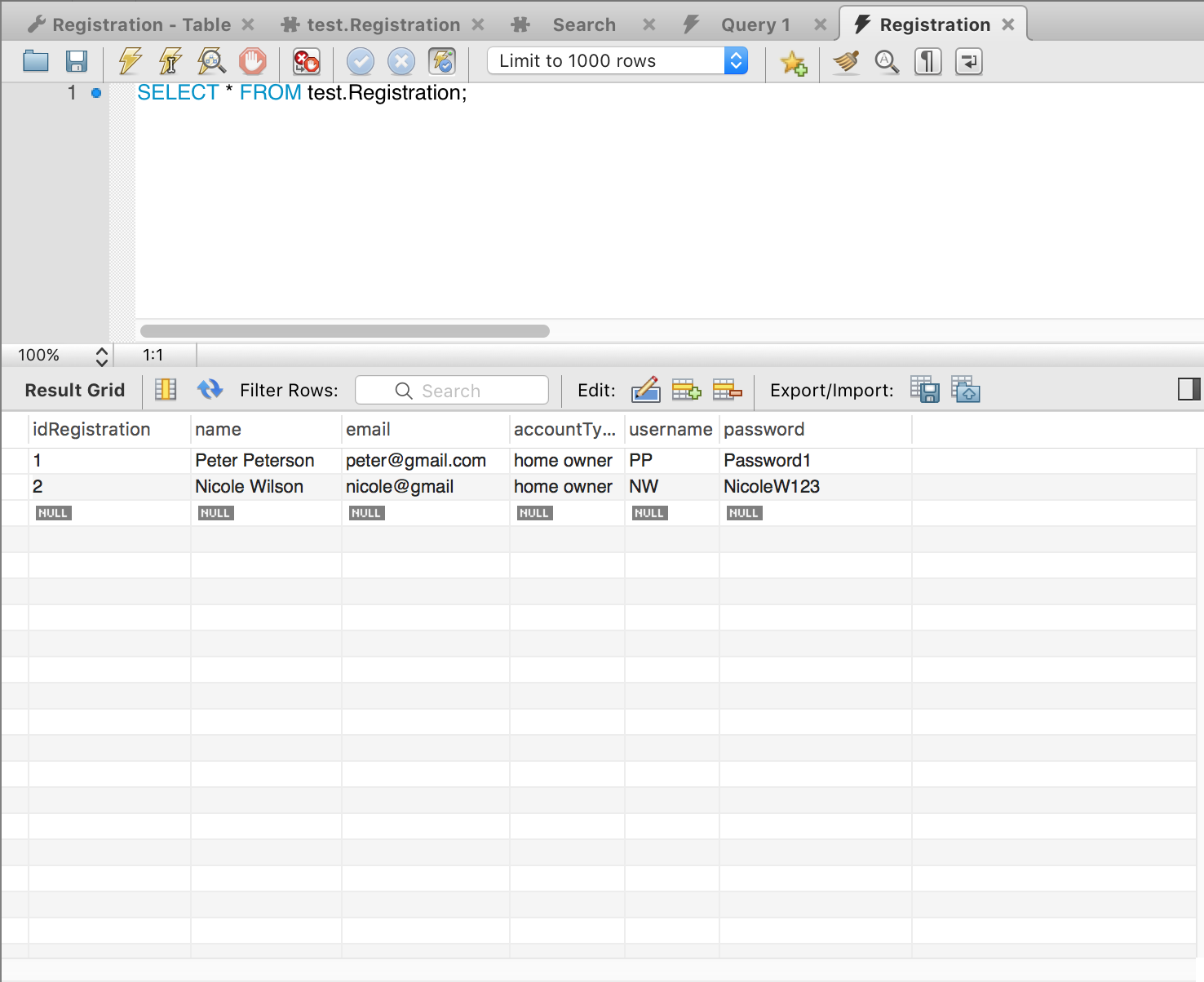


To erase session data the unset function is used. It will then redirect you to login.php page.

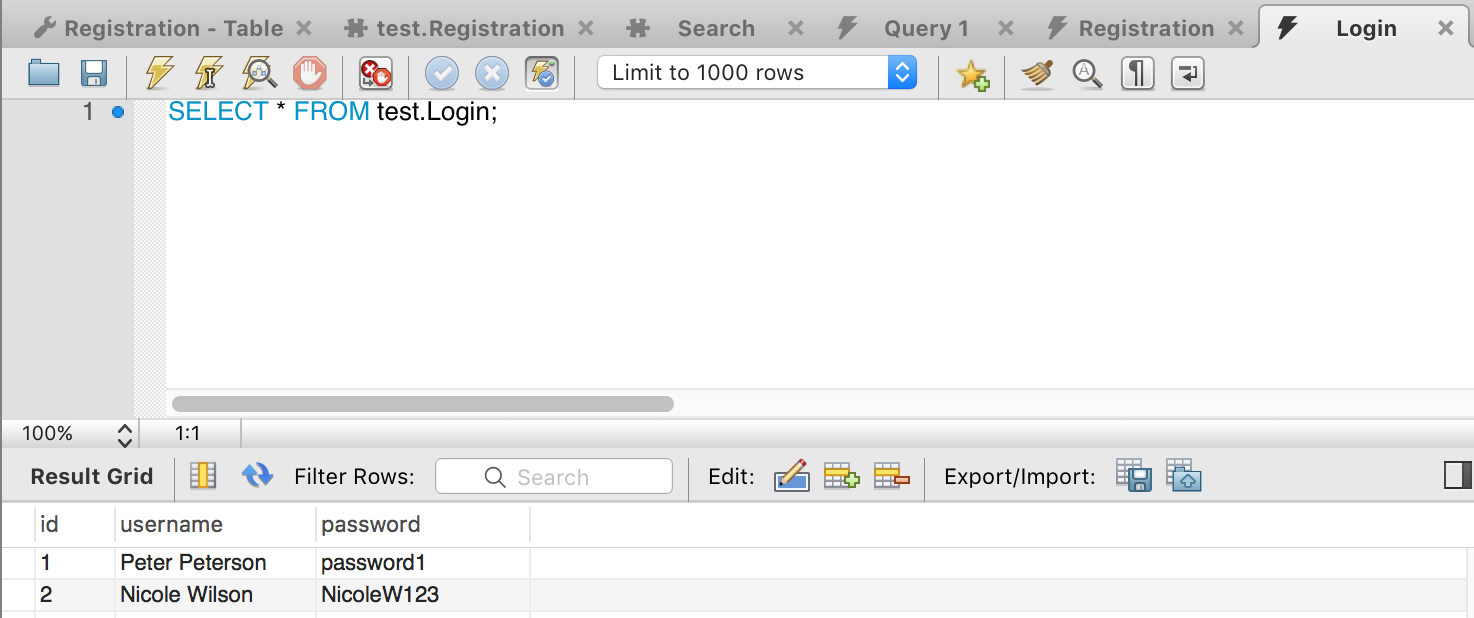


## Database Requirements Specifications

As the development team was using PHP to capture fields and then apply them to the database I decided to use MySQL. Firstly, I created the Registration table to capture a user’s details when they register and account. An example of how the database would store a user’s details can be seen in the figure below.



Secondly, I created a Login table which would retrieve the details which had been applied when a user created an account. The screen shot below is an example of how the database structure was formalised to retrieve information that had already been created. The idea was that the Login table in the database will dynamically store a user’s details once they have created an account from registration form. However, this task was never properly achieved due to the lack of experience from the development team.



## Re-formatted code of webpages to keep website constant

From the start of this project, I elected to use bootstrap template in order for us to cut down on coding time. However, the other developers in my team we not use to this format and had only basic HTML and CSS skills. This required me to re-format the code they had written and structure it to the standard format. The screen shots below give an example of what the website looked like before and after.

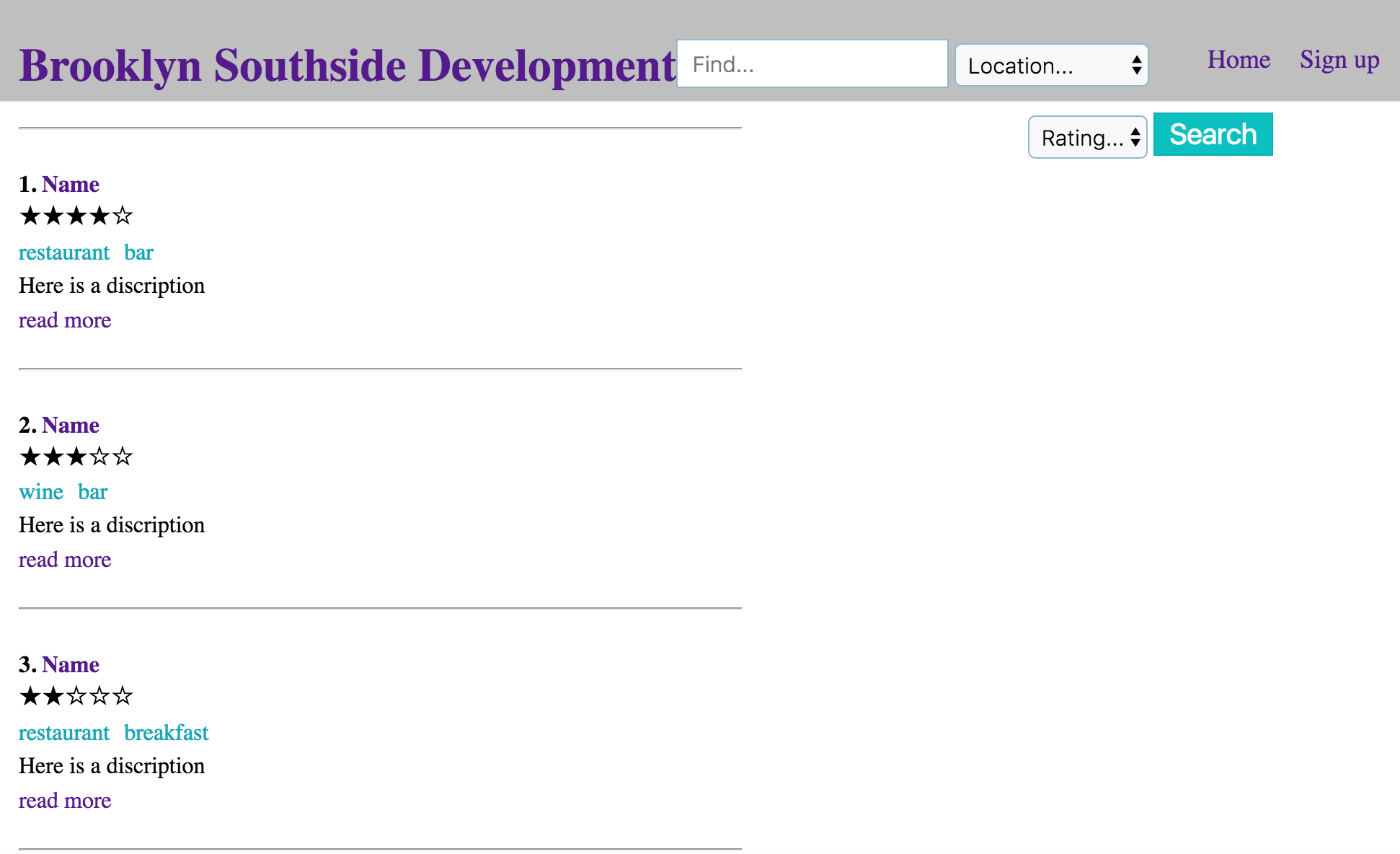




Figure 8 - Search bar (Before)

Figure 9 - Search page resulting (Before)

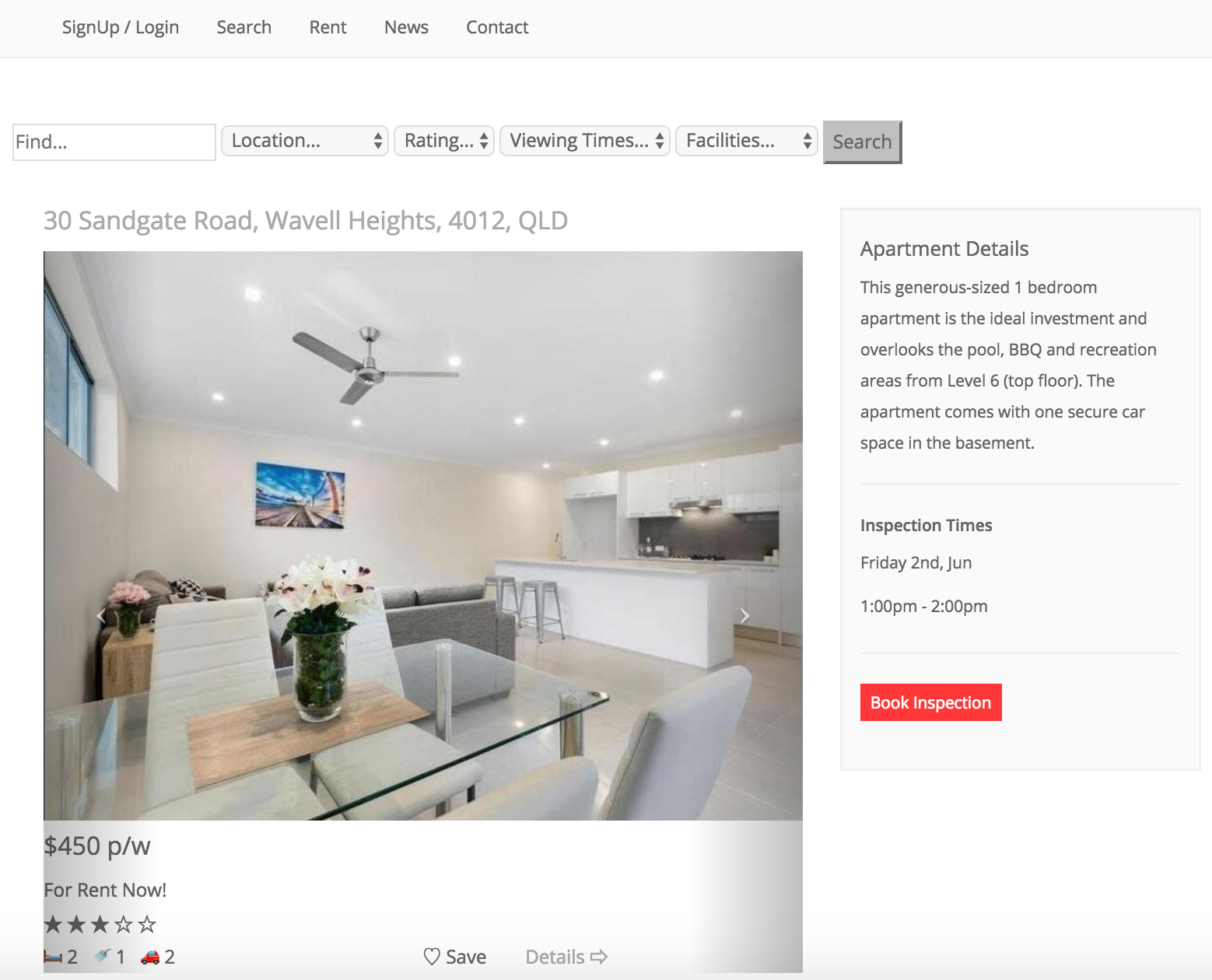
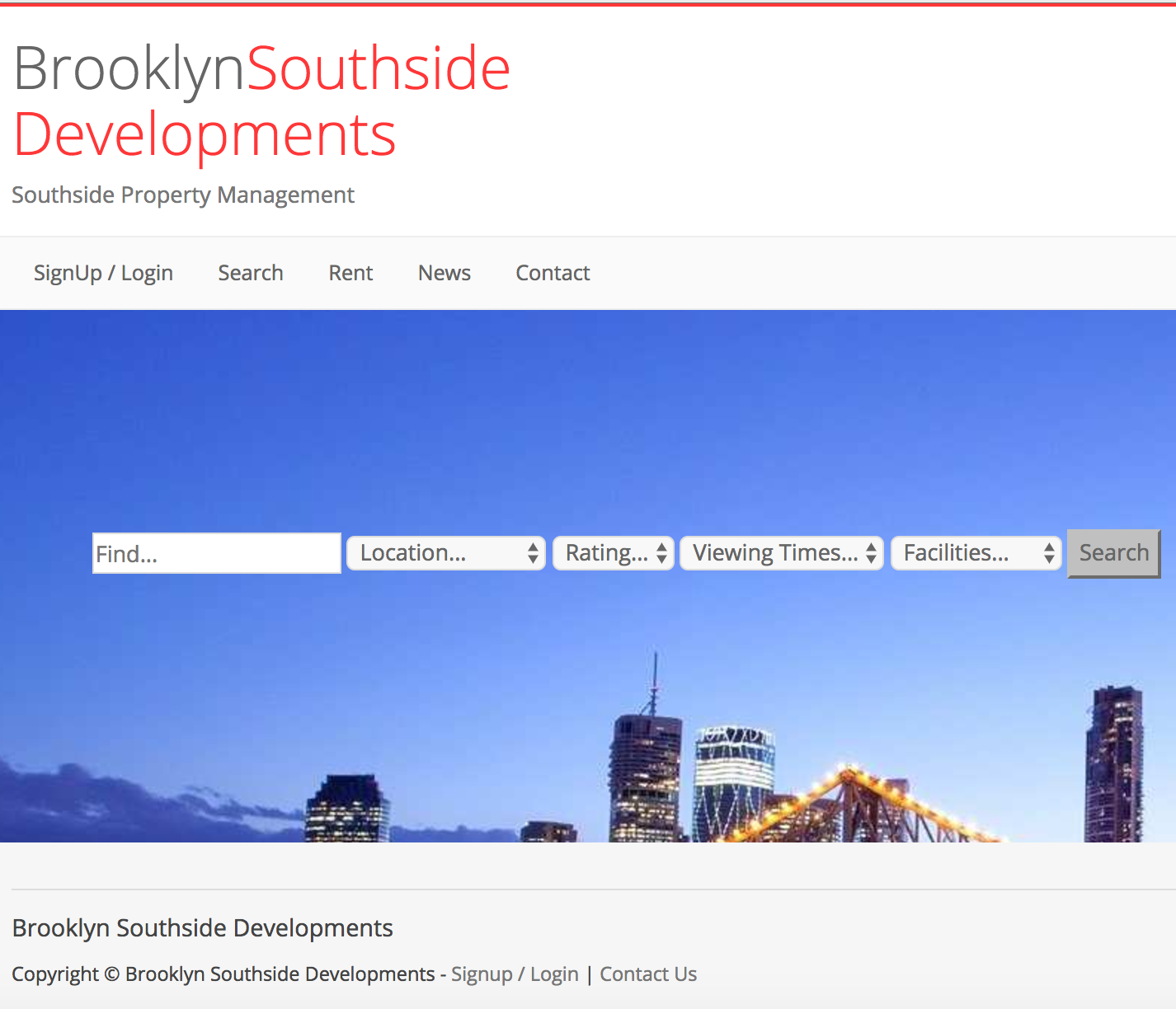


Figure 10 - Search bar (After) Figure 11 - Search Page Resulting (After)

Reformatting the code allowed our website to remain more consistent with the applied theme colours and overall usability. This task was finished in Sprint 2 as seen in the table below. I also ensure the website remained responsive, however, was not able to capture the advertised property details in the database. This task will be finalised in Release 2 allowing the developers more time to research an effective way to keep the website dynamic.

